



DAY 03

A. SUMDIST

```
void DFS(int u, int w) {
    f[u] = 0;
    s[u] = 1;
    for(int v : adj[u])
        if (v != w) {
            DFS(v, u);
            f[u] += f[v] + s[v];
            s[u] += s[v];
        }
}

void DFS2(int u, int w) {
    if (w == 0)
        g[u] = 0;
    else
        g[u] = g[w] + (f[w] - f[u] - s[u]) + (n - s[u]);
    for(int v : adj[u])
        if (v != w)
            DFS2(v, u);
}
```

B. MATCHING

int f[maxn]; // So canh toi da duoc chon trong cay con goc u, u co tham gia 1 canh
int g[maxn]; // So canh toi da duoc chon trong cac con cua u, u khong tham gia canh nao

```
void DFS(int u, int dad) {
    g[u] = 0;
    int delta = 0;
    int gmax = 0, tong = 0;
    for(int v : adj[u])
        if (v != dad) {
            DFS(v, u);
            g[u] += max(f[v], g[v]);
            delta = max(delta, g[v] + 1 - max(f[v], g[v]));
        }
    f[u] = g[u] + delta;
}
```

C. RUNAWAY

+) Hàm tìm tổ tiên thấp nhất có khoảng cách $> L$

```
int ancestor(int u, int L) {
    for(int k=log2(n)+1; k>=0; --k) {
        if (d[u]-d[par[u][k]]<=L) u=par[u][k];
    }
    return u;
}
```



D. DWNSPEED

File Dinitz đi kèm

E. STARTUP

```
struct ST {
    int gmax[4 * maxn];

    void update(int u, int val, int r = 1, int lo = 1, int hi = n) {
        if (lo == hi) {
            gmax[r] = val;
            return;
        }
        int mid = (lo + hi) / 2;
        if (u <= mid)
            update(u, val, 2 * r, lo, mid);
        else
            update(u, val, 2 * r + 1, mid + 1, hi);
        gmax[r] = max(gmax[2 * r], gmax[2 * r + 1]);
    }

    // Tim chi so i lon nhat de max trong doan [i,R]>=x
    int FLeft(int L, int R, int x, int r = 1, int lo = 1, int hi = n) {
        if (R < lo || L > hi)
            return 0;
        if (L <= lo && hi <= R && gmax[r] < x)
            return 0;
        if (lo == hi)
            return lo;
        int mid = (lo + hi) / 2;
        int ret = FLeft(L, R, x, 2 * r + 1, mid + 1, hi);
        if (ret)
            return ret;
        else
            return FLeft(L, R, x, 2 * r, lo, mid);
    }

    // Tim chi so i nho nhat de max trong doan [L,i]>=x
    int FRight(int L, int R, int x, int r = 1, int lo = 1, int hi = n) {
        if (R < lo || L > hi)
            return 0;
        if (L <= lo && hi <= R && gmax[r] < x)
            return 0;
        if (lo == hi)
            return lo;
        int mid = (lo + hi) / 2;
        int ret = FRight(L, R, x, 2 * r, lo, mid);
        if (ret)
            return ret;
        else
            return FRight(L, R, x, 2 * r + 1, mid + 1, hi);
    }
} T;
```